Defining and Using Ontologies as Input for Game Software Factories

André W. B. Furtado André L. M. Santos

Federal University of Pernambuco, Informatics Center, Brazil

Abstract

This short paper details the ontology used to define the product line of the SharpLudus software factory [Furtado 2006a, Furtado 2006b], which is targeted at 2D adventure games. The final intention is to illustrate how ontology research, as input to game software factories, helps to establish common terminology and provide a better understanding of the domain, therefore improving game development industrialization approaches.

Keywords: ontology, digital games, software factories.

Authors' contact:
{awbf,alms}@cin.ufpe.br

1. SharpLudus Ontology

The resulting ontology for the SharpLudus product line is detailed in the following subsections. Due to space constraints, not all of the concepts are explained in detail and we remind that the concepts graph can still be further expanded to present deeper concepts.

1.1 Root Concepts

The top-level concept of the ontology is the *SharpLudus game* concept. The word "game" was not used alone since it can refer to games in general, including games that do not belong to the SharpLudus product line. As shown in Figure 1, a SharpLudus game is associated to six main concepts: configuration, graphics, entities, events, flow and audio.

1.2 Configuration

The *configuration* concept (Figure 2) refers to the main properties of a SharpLudus game, such as its display mode (full screen or windowed), its coordinate system (a convention for defining how the X and Y axes behave, in order to define each point unambiguously in the game), its description (the purpose of the game, its goals, story and so on), its high scores table and screen resolution.

1.3 Graphics

The *graphics* concept (Figure 3) gathers rendering concepts of a SharpLudus game, such as its sprites (animations) and any isolated textual or graphical information that can be displayed to the player. It also includes the heads-up display (HUD) concept, which provides to the game player information about the main character and the game in general, such as the lives indicator, health indicator, time indicator and son on.

1.4 Entities

The *entities* concept (Figure 4) is the base unit of a SharpLudus game design. An entity is anything that can react with anything else in any way. It can be an ordinary item, such as a projectile or something to be collected, the main game character or a non-player character (NPC). It is important to notice that some concepts, such as the main character item inventory, are exclusively related to the product line domain (based on adventure games) and may not apply to other game genres.

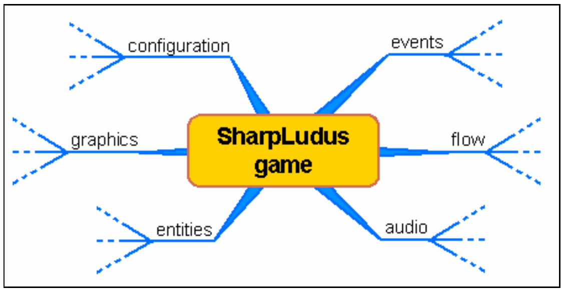

Figure 1: Root concepts

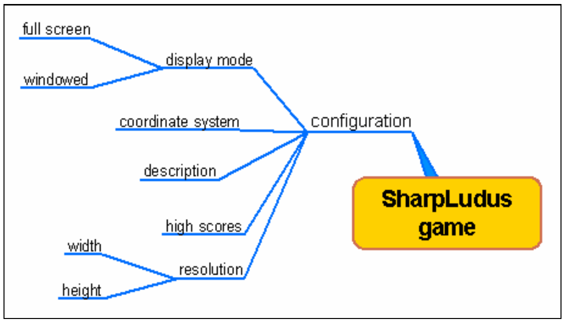

Figure 2: Configuration concept

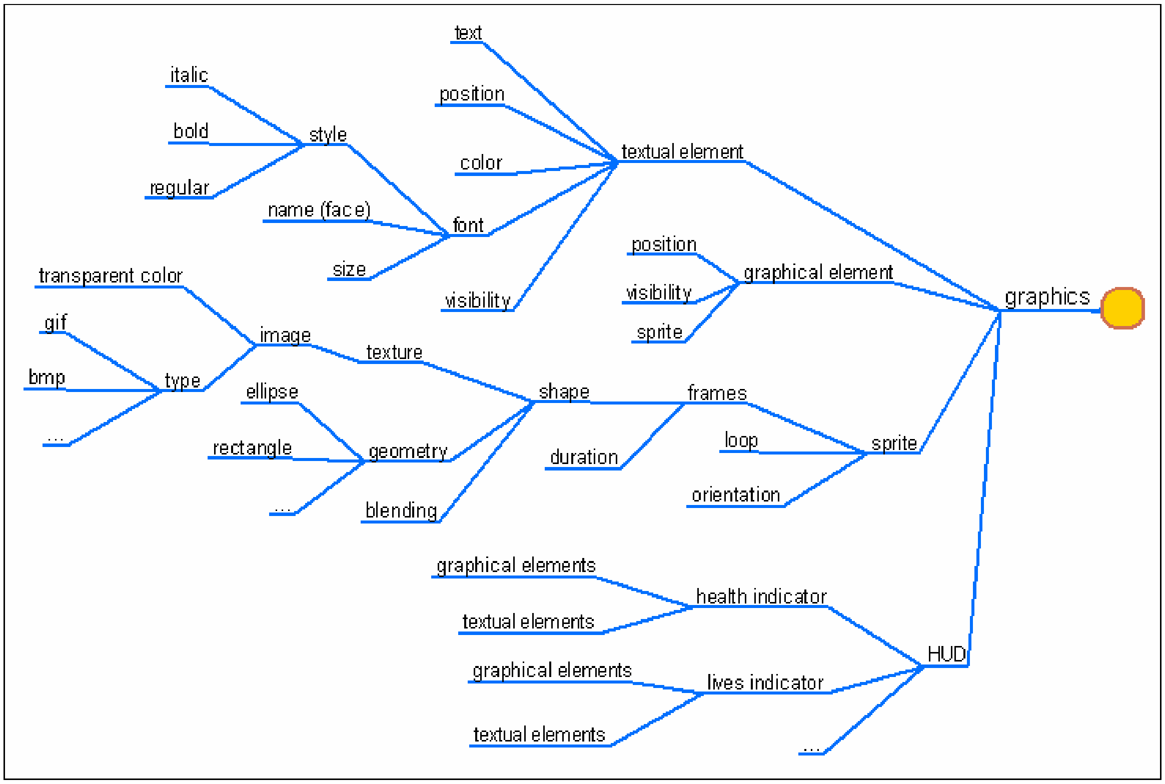

Figure 3: Graphics concept

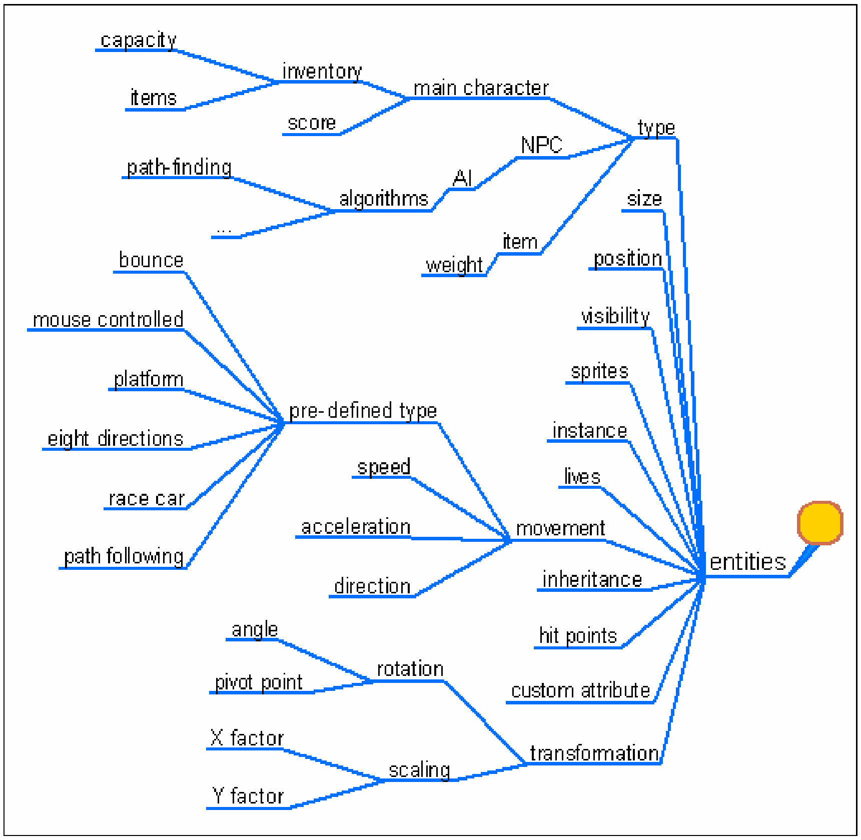

Figure 4: Entities concept

1.5 Events

Events (Figure 5) are special conditions that occur to a SharpLudus game, fired by one or more triggers, and that cause one or more reactions. Once more, triggers and reactions can be specific to the domain, such as "set current game room" or "add item to inventory".

1.6 Flow

The *flow* concept (Figure 6) represents a SharpLudus game behavior by means of a succession of "game states", which can be information display screens or world rooms. Such rooms are interconnected and typically represent the spaces where the main character and NPCs walk around, and where items await to be collected. However, the name "room" does not necessarily imply a closed space, since it is only used as an intuitive way to name the subdivisions of the game world.

1.7 Audio

The *audio* concept (Figure 7) represents every sound that can be reproduced in a SharpLudus game, such as sound effects and background music, as well as some related properties.

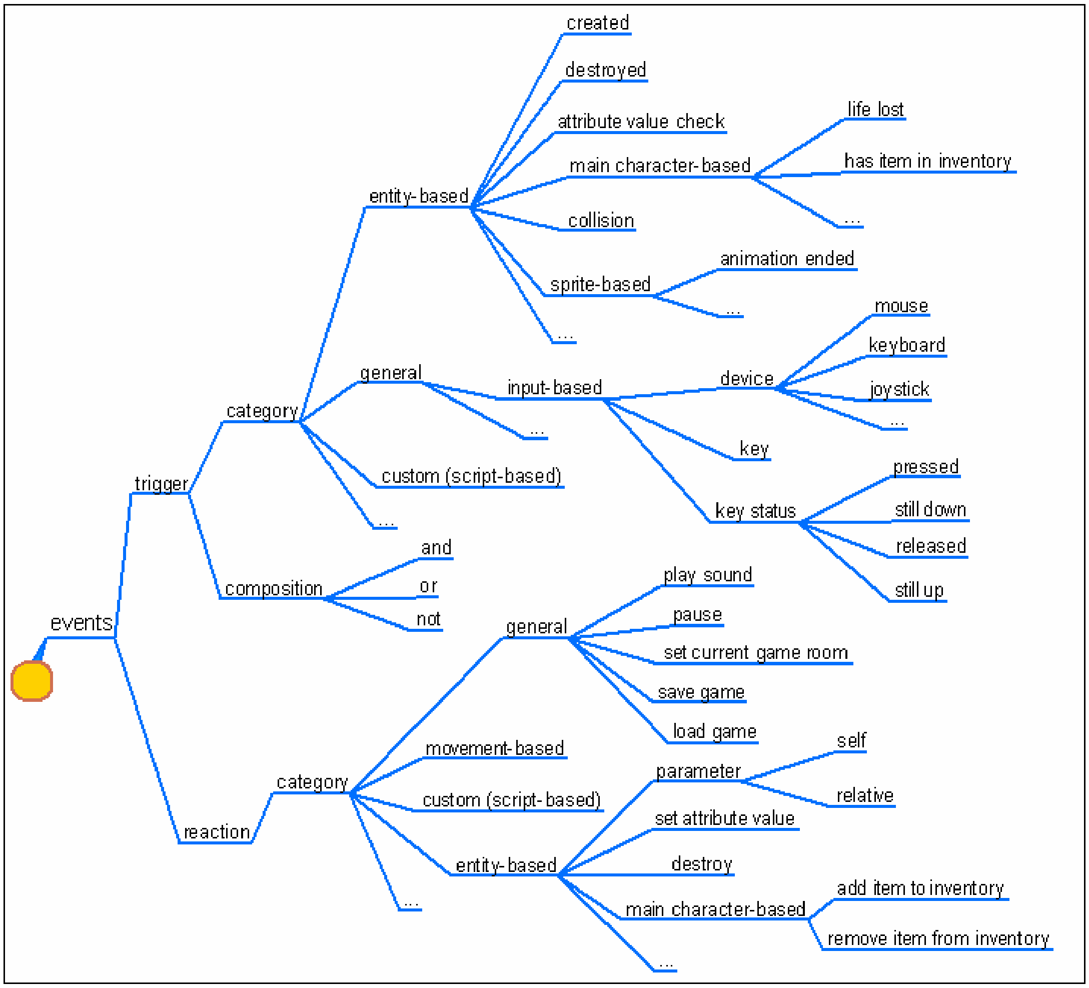

Figure 5: Events concept

2. From Ontologies to Factory Assets

Once an ontology focused on the target factory product line is created, it is possible to use it as the basis for creating factory assets aimed at improving the experience of game developers and designers.

In the SharpLudus software factory, the ontology previously presented was used to create a visual domain-specific language (DSL) named SLGML (SharpLudus Game Modeling Language), through which game designers can intuitively specify part of the game world.

Its concepts and relationships were inherited from the ontology, while a visual representation (syntax) was defined for the DSL. Figure 8 presents a modeling experience with SLGML.

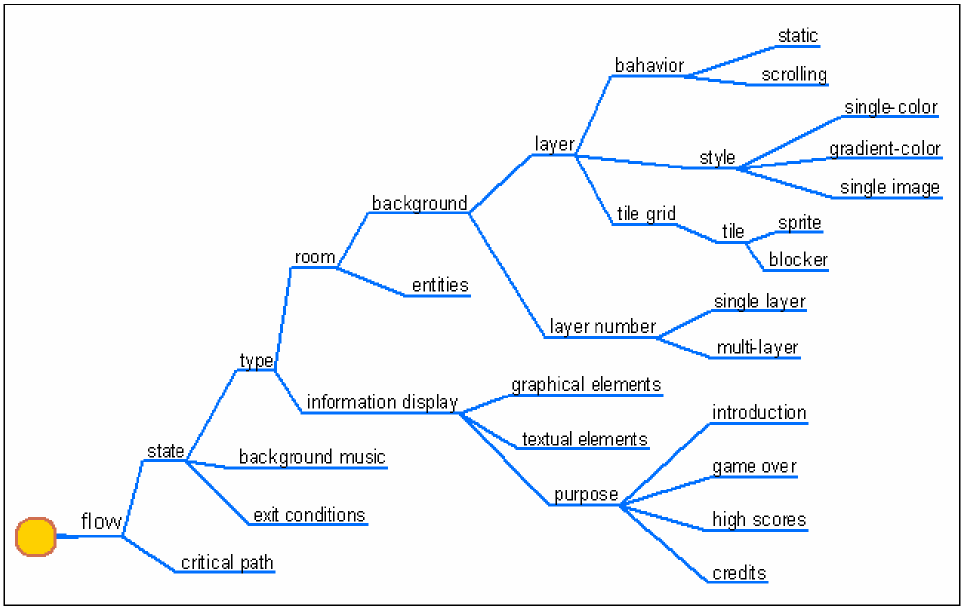

Figure 6: Flow concept

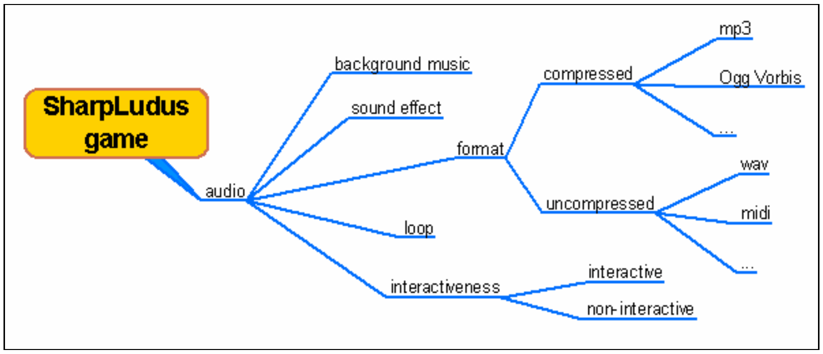

Figure 7: Audio concept

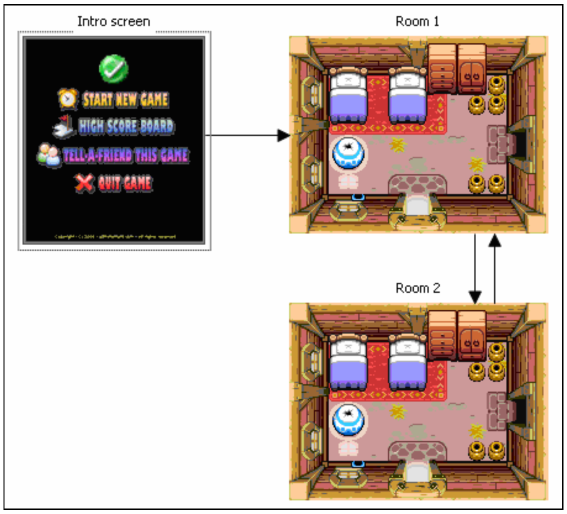

Figure 8: SLGML modeling experience

3. Conclusion

This paper detailed an ontology targeted at 2D adventure games, which was used as input for the SharpLudus game software factory. The ontology played a key role in defining one of the most important factory assets: a domain-specific language used to model the flow and behavior of the game.

The results obtained so far empirically shows that this is indeed an interesting approach. However, it is worth noticing that the presented proposal alone do not ensure the success of a game development. In fact, no technology is substitute for creativity and a good game design. Game industrialization, ontologies, languages, frameworks and tools are means, not goals, targeted at the final purpose of making people have entertainment, fun and enjoy themselves. Players, not the game or its constituent technologies, should be the final focus of every new game development endeavor.

References

FURTADO, A.W.B., 2006. *SharpLudus: Improving Game Development Experience through Software Factories and Domain-Specific Languages*. MSc thesis, Federal University of Pernambuco.

FURTADO, A.W.B., 2006. *The SharpLudus Project* [online] Federal University of Pernambuco. Available from: www.cin.ufpe.br/~sharpludus [Accessed 30 September 2006].